More Than A Name

Dilwyn Jones Computing

September/October
1992

DILWYN JONES COMPUTING

41 BRO EMRYS, TAL-Y-BONT, BANGOR, GWYNEDD, GREAT BRITAIN, LL57 3 YT
TELEPHONE: 0248-354023

WE SELL A WIDE RANGE OF SINCLAIR QL SOFTWARE, PROGRAMS WE PUBLISH OURSELVES AND THIRD PARTY SOFTWARE. MOST ITEMS ARE HELD IN STOCK WHERE POSSIBLE, BUT WE WILL OF COURSE ORDER ANY ITEM NOT CURRENTLY IN STOCK AND QUOTE YOU A DELIVERY TIME.

CATALOGUE ASK FOR A COPY OF OUR QL PRODUCTS CATALOGUE, FREE ON REQUEST. IT CONTAINS MORE DETAILS OF OUR SOFTWARE AND SOME SCREEN PICTURES FROM THE PROGRAMS.

SOME OF OUR SOFTWARE IS AVAILABLE FROM EMSOFT AND MECHANICAL AFFINITY IN THE U.S.A., QITALY CLUB IN ITALY, JURGEN FALKENBURG AND QLYMPIC COMPUTER SYSTEMS IN GERMANY. OR YOU CAN ORDER DIRECT FROM US, OF COURSE. IF CONTACTING US BY TELEPHONE, REMEMBER THE UK IS 5 TO 8 HOURS AHEAD OF THE U.S.A. WE HAVE AN ANSWERING MACHINE FOR WHEN WE CAN'T ANSWER IN PERSON AND WE DO TRY TO CALL YOU BACK IF REQUIRED.

WE CAN ACCEPT ORDERS BY TELEPHONE, PAID FOR BY CREDIT CARD FOR LOW EXCHANGE COSTS AND QUICK DELIVERY. WHY NOT CLUB TOGETHER TO BUY PROGRAMS AT REDUCED COSTS AND TAKE ADVANTAGE OF QUANTITY DISCOUNTS (5% DISCOUNT FOR 2 PROGRAMS, 10% FOR THREE OR MORE). PLEASE ADD 2.00 POUNDS STERLING PER PROGRAM ON MICRODRIVE CARTRIDGE.

PAYMENT BY CREDIT CARD (ACCESS, VISA, MASTERCARD, EUROCARD, CONNECT), EUROCHEQUE (IN POUNDS STERLING), INTERNATIONAL POSTAL/MONEY ORDER, CASH, CHEQUE IN POUNDS STERLING DRAWN ON UK BRANCH OF A BANK OR BUILDING SOCIETY. PRICES IN THIS ADVERT ARE IN UK POUNDS STERLING CURRENCY.

POSTAGE SOFTWARE POST FREE IN THE UK, ADD 1 POUND PER PROGRAM TO OTHER COUNTRIES

FILE TRANSFER
DISCOVER	20.00
MULTI DISCOVER	30.00
TEXTIDY	15.00

LEISURE
SOLITAIRE	15.00
CRICKET SECRETARY	12.00
QUESTION MASTER	10.00
FLEET TACTICAL COMMAND 2	49.95
FTC DATA PRINTER	9.95
THE FUGITIVE	9.95
COCTAILS WAITER	10.00

PROGRAMMING
BASIC REPORTER	10.00
BUDGET QLIBERATOR	25.00
QLIBERATOR 3.35	50.00
QLOAD and QREF	10.00
DISA disassembler	29.00
EASYPTR II	49.00
BUDGET EASYPTR II	35.00

FILE HANDLING
FILES 2	12.00
FILEMASTER	12.00
THE GOPHER	12.00
WINBACK V2.01	25.00

D.T.P.
PAGE DESIGNER 2+	40.00
PD2 UPGRADE	20.00

DISPLAY SOFTWARE
VISION MIXER 1	10.00
VISION MIXER PLUS	22.50
UPGRADE VM1 TO VM+	14.50
PICTUREMASTER	15.00
PICTUREMASTER PLUS	20.00
UPGRADE TO PM PLUS	5.00

GRAPHICS
THE PAINTER 4.04	25.00
(HP Deskjet printer option)	
THE CLIPART	12.00
QRACTAL	20.00
DEMO QRACTAL SCREENS	5.00
IMAGE PROCESSOR V2	15.00
UPGRADE TO VERSION 2	10.00
PD2 CLIPART SCREENS	10.00
SCREEN SNATCHER	10.00
TEXT 'N' GRAPHIX	20.00
TRANS24 converter	10.00

TEXT
KING JAMES BIBLE TEXT	
PLAIN TEXT FILES	20.00
QUILL WP" DOC" FILES	20.00
SPELLBOUND	30.00
SPELLBOUND SE	50.00
SPELLBOUND UPGRADE	30.00
QUICK POSTERS	10.00
ROB ROY SOFTWARE PACK	10.00

DATABASE
ADDRESS LABEL PRINTER	15.00
DATA DESIGN	50.00
FLASHBACK	25.00
FLASHBACK SE	40.00
UPGRADE TO SE VERSION	20.00
GENEALOGIST	19.50
GENEALOGIST V2	30.00
BUDGET GENEALOGIST	12.00
SUPER DISK INDEXER	12.00
DBEASY	15.00
DBPROGS	15.00

OTHER SOFTWARE
HOME BUDGET (UK)	20.00
REMIND ME	12.00
REMIND ME PLUS	20.00
SCREEN ECONOMISER	10.00
SLOWGOLD slow down	5.00
TASKMASTER	25.00
SUPER DISC LABELLER	10.00
THE CAT	5.00
MEGA-TOOLKIT	ask!
(ready during the summer)	
MINI PROCESS CONTROLLER	
SOFTWARE UTILITY	9.95
NETWORK PROVER BOX	3.50

LABELS
100 disk labels	2.50
100 m'drive labels	2.00
100 address labels	2.00

APPROXIMATE US DOLLARS EXCHANGE RATE IS 1 POUND = US $1.85

EDITORIAL
NEWPORT, RHODE ISLAND, USA - THE EDITORIAL STAFF

With this issue, we begin a new feature entitled "MORE THAN A NAME". Along with a picture on the front cover, will be an auto-biography. We intend to highlight those people/companies that you have heard of over the years, but probably know very little about. These are people who have, and are contributing much to the furtherance of QDOS and the QL.

Dilwyn Jones of DJ COMPUTING is the first of many we hope to feature in this manner. We have already received commitments from Jochen Merz of Jochen Merz Software, Freddy Vaccha of Digital Precision, and Stuart Honeyball of Miracle Systems.

Continuing in the same vain, we intend to highlight User Groups from around the world. In our next issue, the QLEA sub-group of Quanta will be featured. If you'd like to see YOUR group in IQLR, we need a one or two page article, (although size is not critical) describing your group and activities. A picture of the group would be welcome, (black and white photo) but is not required.

As the fall and winter computing seasons approach, so does the promise of some exciting new products, especially some long awaited ones from Miracle Systems.

Along these lines, a new product, the "ROM SWITCH" from QLEA (a Quanta sub-group) has arrived. It allows you to run both Minerva Mark I and a standard set of ROMs in the same machine (note the review and their advert in this issue). We've had it installed for about a month now, and can heartily recommend it.

We've also received the revised version of Miracle Systems Serial to Centronics printer interface and is it slick, no more bulky box on the back of your printer; nice, neat, and clean, I can't figure out where they put the hardware (see their advert for additional information).

OOPS !! In our last issue, we reported that Ron Dunnett was elected Chairperson of QUANTA, in reality, Ron was elected General Secretary, Bill Newell is the new Chairperson, he also retains the post of Membership Secretary. Our apologies to both Ron and Bill.

Frequently, readers call or write informing us of bugs in programs or other problems. We deal with these by notifying the dealer or author in question. Following, are the results of our latest inquires.

It appears that there may be problems with early versions of the SERMOUSE software. Jochen Merz has informed us that defective copies should be returned either to him, or Albin Hessler the author (see documentation for addresses) for an upgrade.

Miracle Systems asks that all bugs found in the Gold Card ROM be reported to them, so they may be passed on to Tony Tebby for correction in the next version ROM. Miracle offers FREE ROM upgrades as they become available (current ROM version is 2.28).

Also from Miracle, comes a request to inform you of a misconception concerning the new ROM for The Trump Card. The new ROM allows users to connect the 3.2Meg. disk drives to their Trump Cards, BUT THEY WILL ONLY OPERATE IN 720K MODE.

EDITORIAL (cont'd)

They will NOT operate in either 1.44Meg or 3.2Meg mode. The Trump Card circuitry doesn't allow for High or Extra High Density.

An incompatibility problem between MINERVA and the game Solitaire can easily be fixed by applying the QLIB bodge supplied with MINERVA. Peter Hale of EMSOFT, the distributors of Solitaire, also stated that it runs fine without the bodge in a multi-tasking environment such as Taskmaster.

FOR SALE

1 - Complete North American Sinclair QL with original software and documentation. Also the games "The Lands of Havoc" and "Knight Flight". A Wico brand joystick is also included. For pricing or information, contact: Mrs. Rita Timko - 26264 Aaron Avenue - Euclid, OH 44132 - USA or call 216 261 0471.

DISK DRIVE CABLE EXTENDER
NEWPORT, RHODE ISLAND, USA - SEACOAST SERVICES
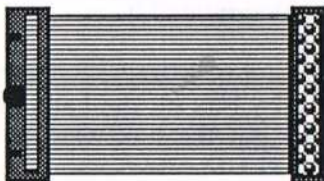
DISK DRIVE EXTENDER

The DISK DRIVE CABLE EXTENDER is a product that came about out of necessity. After hasselling with the tight cable and cramped working conditions when changing drive cables on the Gold Card (works equally well on the Trump Card), and frequently pulling the Gold Card loose from the edge connector, we decided that what was needed was a small two-way cable that would extend the connector far enough to make it easy to work with and also reduce wear and tear on the GC socket.

After learning that we weren't the only ones with this problem, and having a number of requests to make one, we decided to offer it to our readers.

The DISK DRIVE CABLE EXTENDER (DDCE) is about three inches long with a header on one end, and a 34 pin IDC socket on the other (see illustration). All components are high quality.

The price of the DDCE is $10.00 US FUNDS plus $3.00 S & H in North America, and $10.00 plus $5.00 S & H for the rest of the world. We will also except the equivalent in DM or Pounds Sterling.

Send your orders to: SeaCoast Services - 15 Kilburn Court - Newport, RI 02840 - USA.

A Quantum Leap in QL Wordprocessing

We are proud to present our new state-of-the-art wordprocessor, text87plus4. After a long period of development leading to its first release, and another four months spent incorporating users' suggestions, the definitive, optimised version 3 of plus4 is ready.

plus4 is not just an improved version of the original text87; It is a complete rewrite from scratch. Recent technology advances have allowed us to develop a program which is MILES ahead of any QL application. We have kept the technology of the original text87, including our state-of-the-art series of printer drivers which exceed the capabilities of the latest PC wordprocessors.

text^{87} plus4

+1 USER FRIENDLY TO THE EXTREME

You will hardly ever need our new well-written manual. An automatic setup and installation program allows you to select a suitable driver for your printer and copies all the necessary files to your disk. Run plus4 and a menu allows you to load a file or start a new one. An extra line of instructions and another line containing the current setting are displayed. Press <F1> and a window offers more help related to the menu options (context-sensitive). If you select Load you do not have to remember the file name; just press <UP> or <DOWN> for a list. Use the same keys to select the file that you wish and press <ENTER>.

This user-friendly command system governs the program in every area. Extensive context-sensitive help is only an <F1> away. No need to type in file-names, etc. if the program can offer a list in a selector box. Commands and key-presses are highly compatible with those used in Quill and function keys perform the same operations.

+2 THE MOST POWERFUL QL WP

Plus4 provides all the navigation and editing facilities you would expect and a lot more. Extensive editing facilities include cursor move (by character, word, line, paragraph, screen, page) erase (by character, word, line) block operations (copy, move, delete) goto (line, page, top, bottom, section, block). Insert and overwrite modes. Very fast search and replace backwards and forwards, case dependent and independent. Special characters include hard-space, hyphenation, hard and soft hyphens. In operation plus4 reformats the text as you edit and preserves the format of each paragraph no matter how many different formats you use in your text. Everything is automatic.

As a Quill user you would naturally expect your wordprocessor to remember different tab and margin settings for a document. You would expect to freely add to old texts without having to bother about those settings over and over again. Not surprisingly, text87 is the only other QL program that supports this important, user friendly feature of Quill.

File Operations include load, save, merge, block save (in plain ASCII or as fully formatted document) import (Quill files retaining bold, underlined, etc. or any other file, including those exported from Archive and Abacus or from other programs). The combination of all these powerful commands enables you to move text from one document to another effortlessly.

Integrated Spell Checker displays selector boxes for browsing the dictionary and automatic replacement of the selected word. This is automatically capitalised if the original began with a capital. Choose between large (over 210,000 words) and small English dictionaries or French or German (all supplied with the program). You can add any word in your text to the dictionary by just pressing a key. Your word lists can be saved and loaded at will or added to the dictionary on a permanent basis. (You can actually edit the dictionaries to your requirements).

Multi-Window Multi-Document plus4 goes far beyond multi-tasking. With one copy of plus4 up to 8 document windows can be open simultaneously. Up to 8 files can be on screen and more than one window can be open over a document so that you can edit the text while looking at a different part of it or at a different document. Resize, Zoom, Tile and Stack commands allow you to arrange the windows manually or automatically and switch instantly between them.

Page-Preview and Pagination Page and column ends are constantly displayed on the screen. plus4 takes into account all the changes of line spacing (you can fine tune the line spacing in different parts of the text between 0 and more than 1 inch). The Page preview command shows your text in full A4 (and other size) pages. Each word is represented by a rectangle, giving a realistic picture of the printed page before you commit the text to paper. This command alone will save you a lot of time and effort.

+3 UNRIVALLED PRINT QUALITY

In text and character formatting, text87plus4 is miles ahead of the so-called competition. Simply, no other QL program can produce similar results. Used with the appropriate printer-driver, text87plus4 can utilise the different founts and character sizes built into modern printers. It fully supports proportional spacing (such as used for this text) and justifies correctly. You can use any combination of small and large founts on the same line and be assured of a perfect printed result. You can set up multiple paragraph formats with different margins and line-spacing for each. You can use any combination of ordinary tabs with right, centre and decimal tabs in each line of text. You can also format the page the way you want, using several columns plus headers and footers. For desktop publishing, you can use several different page layouts in the same document.

WYSIWYG (what you see is what you get--pronounced wizzy-wig) Years ago this word referred to the absence of printer control codes from the screen. It is now used to distinguish word-processors which display different amounts of line-spacing and different character sizes and styles (e.g. double width, proportional). text87plus4 is the only QL wordprocessor that can be called WYSIWYG by current standards.

+4 FASTEST QL WORDPROCESSOR

Figures speak for themselves. We tested text87plus4 on a QL with memory expansion and disk-drive and on an Atari ST with QL emulator. A 70 page text of over 24,000 words and 141,000 characters was used for these tests. Load document: 25s (STQL 17s). Save document: 37s (STQL 32s). Automatic search and replace (includes automatic reformat of modified text) 580 instances: 43s (STQL 14s). Change justification from full justified to left justified or back: less than 2s (STQL 1s). Change right margin from 66 to 72 units and reformat whole document: 65s (STQL 17s). Move block of 10 pages from top to bottom (including manual marking and positioning): 35s (STQL 15s). Scroll the whole screen over text line-by-line (either up or down) 100 lines: 19s (STQL 5s).

plus4 is supplied with over 30 ready-made printer drivers supporting 9pin and daisywheel printers. Extra drivers for 24pin, Bubblejet, Deskjet and laser printers support the resident letter-quality founts built into the printer. All our drivers come with predefined translates for QL's extended character set.

plus4 is fully compatible with all QL roms. Gold Card, ST QL, etc. Requires disk drive and 256K memory.

Prices (inclusive of Air Mail to overseas)

text87plus4	£ 79.00
upgrade to **plus4** from v. 3.00 (last chance)	£ 39.00
2488 drivers for 24pin and Bubblejet printers	£ 19.00
typeset90-deskjet drivers for all HP Deskjets	£ 19.00
typeset90-Epson GQ3500/5000, EPL4100/7100 lasers	£ 39.00
fountext88 + founted89	£ 39.00
Graphic driver for 9 & 24-pin printers with over 30 founts	

UPDATE! MAGAZINE

QL

Z88

TS2068

QL

Z88

TS2068

QUARTERLY MAGAZINE - $18yr
P.O.BOX 1095, PERU,IN 46970
SINCLAIR NEWS IN NORTH AMERICA

DILWYN JONES COMPUTING
A POTTED HISTORY OF MY COMPANY AND I!

My involvement with Sinclair computers goes back to the very early days when I wrote some books on programming the ZX81/TS1000 and Spectrum/TS2068.

After writing a great deal of BASIC programs in the first few years of the QL, I eventually wrote what were my first large programs, Page Designer 2, which was published by Sector Software after I met David Batty at a Quanta QL workshop. Page Designer 2 was a much enhanced version of the rather limited and difficult to use original Page Designer program in the Quanta library.

Following my involvement with Sector Software, I spent a year or two not really doing very much commercially for the QL. I kept on contributing programs to the Quanta software library, but that was it until I decided to start my own company to produce software for the QL. I went on a Business Enterprise Programme course to learn about starting the company and then DJC was born!

I started with just three utility programs - BASIC Reporter (a BASIC programming aid), Vision Mixer (a screen advertising utility) and Wordscheck (a word count utility) - and also supplying floppy disks and other little bits and pieces. Those early days were shaky to say the least! Advertising was expensive and limited and it was an uphill struggle just to keep going. I nearly gave up more than once, but selling items at shows gave me the income and enthusiasm to carry on.

The next program I wrote was Quick Posters, which nearly did not get published! It was written to produce simple text only posters for use on my stand at shows and was only meant for my own use. Enough people asked how they had been produced to persuade me to launch it as a commercial program. Considering that it does not work on many types of printer, it has been quite a successful program.

As the business started to expand and I began to advertise more widely, I decided to advertise for prospective software authors in an effort to expand on the number of programs I sold. The first reply came from Joe Haftke, whose first program had been launched by another QL company at the time, but he was looking for another publisher to work with. That became the start of a long association. British QL users will be familiar with the sight of Joe helping me on the stand at some shows in England - his hard work, support and willingness to help at all times deserve a mention here.

The advert which invited QL authors to contact me produced an enormous response. I quickly became bogged down ploughing through the endless pile of disks and microdrives, some without instructions, some of which would only work on one QL ROM version, and some which I could not understand at all!

A great deal was refused, for various reasons. Some rather promising ideas never actually turned into finished products either because the authors got fed up of waiting for me to wade through them all and reply, or because they took offence when I dared to suggest changes to their masterpieces, or because I did not know enough about the field concerned and had to enroll the help of other people to assess the programs, some of whom never did finish the job or gave negative reports on the program.

Some very successful programs came out of all this, though, such as the Super Disk Labeller, which became one of my best selling programs! The same author later produced

D. JONES (cont'd)

another popular program, the Super Disk Index for cataloguing a user's floppy disk collection.

By now I was supplying printed manuals with the software, a policy I have stuck with ever since. As the number of programs grew, it became more and more expensive to get them printed in the number I was selling, so I went and bought a photocopier - an expensive investment but one of the best I have made, even if I do curse it at times, if it breaks down, or seems to take all day to photocopy stocks of large manuals!

Shortly afterwards I met Chris Boutal. Many readers will be familiar with his excellent Genealogist program, now in its Second Edition.

Interest in using the QL for family tree and family history work had been stimulated by an article in QL World and by the Family Tree program by Andy Carmichael. Chris wrote a BASIC program to be used for Genealogy and compiled it with QLiberator, rather than use an established database like Archive. He started to sell it through small adverts.

We met at a Quanta Workshop in Worthing, England, for the first time and were introduced to each other by a Quanta official. I remember swapping a sample copy of Quick Posters for a sample copy of QL Genealogist. Little did I know then how successful it would be! In fact, at the time, I didn't think it would sell very well at all - how wrong I was to be proved! It received excellent reviews and the orders flooded in over the next few weeks after the review came out. That was one of the most chaotic Christmas periods I have ever had! Everyone seemed to want a copy!

It has sold consistently well since those early days and we have since also done a roaring trade in the second edition, which continues to sell well. Apart from the disk labeller program, no other program from the range I publish has generated as much praise as QL Genealogist.

Sadly, some of our range of programs have not done anywhere near as well. Programs such as Question Master, The Gopher and Cocktails Waiter have not sold in very large numbers.

More positively, the range of programs has expanded. As well as publishing DJC programs, I have also established a policy of buying in third party products such as the Di-Ren range, Liberation Software products, Albin Hessler Software, PROGS, Dave Walker and the authors of some programs previously available from Sector Software.

One of the success stories of all this was The Painter. Graphics and printing programs seem to attract a lot of interest on the QL. The Painter is an old program, but it has been re-written several times and is now in its fourth incarnation (with several versions of the fourth incarnation), which means that the facilities have been refined and it has been well debugged. It is good when used from the keyboard, but is fantastic when used with a QIMI mouse (available from Jochen Merz or from Quanta)! If you have never used pointer environment before, it takes a while to get used to it but once you do, you wonder how you ever managed any other way! Certainly, this and DATA Design were the programs which persuaded me to use pointer environment (and QPAC1 and QPAC2) as standard on my QL.

D. JONES (cont'd)

All this rapid expansion caused an unforeseen problem - there was no room in the advert to describe the programs. So I simply put what I could in the adverts and concentrated on producing a catalogue to give more information to customers about the programs. It took some time to put together and is a little expensive to produce, but has been well worth it.

We have been caught out by some unexpected developments on the QL scene. Some of our programs suffered from compatibility problems when used with some versions of the Minerva EPROM, as also happened with the Gold Card. I worked hard with some of the authors to try to improve the situation - at the moment I am getting remarkably few complaints about compatibility problems.

1992 had a difficult start. Apart from the fiasco with the delays on the Page Designer 2 Plus project which I hope will be finished by when this is in print, for some reason I started to receive an incredible amount of queries with orders, or simply just queries. It was very time consuming and, of course, slightly annoying since I have tried to send out goods ordered as quickly as possible after receipt of the order and it meant that anything needing more than a very short reply got delayed until I was able to find time to reply (remember that DJC is not a full time concern). This annoyed many people who were not (until I explained to them, a time consuming exercise in itself) aware of what difficulties long letters and long telephone calls cause me. I also received a large number of queries about software which does not have anything to do with me. It has been quite difficult to persuade customers (and sometimes authors) that all too often I simply do not have the time to go into anything at length - there is always someone or something else waiting in line with demands on my time. I just do what I can.

That, and the problems at QL World, were the difficult part of 1992 so far. On the bright side, there are some very interesting projects in progress at the moment. A Mega Toolkit (for lack of a better name) containing a very large number of BASIC extension is planned. Authors will be able to include a runtime only version of this in commercial software free of charge (apart from the initial purchase price) provided users are told who wrote and sells it. A lot of work has gone into this and I am confident that it will be well received when finished.

A disk sector editor for the various disk densities supported by the Gold Card disk formats is in preparation. Winback version 2 is now out and work continues on DATA Design to develop this excellent programmable database still further.

One author is currently working on a utility to convert Archive language programs to DATA Design language applications. What this means is that, if you have QLiberator, for example, you may be able to produce what is, in many respects, an Archive database compiler! At the simplest level, it will enable you to easily move from Archive to DATA Design. Work on this is still in its early stages and it is too early to say for certain even if the project can be completed or what form it will finally take.

Another author is working on a banner and poster making program which uses outline fonts for better quality text printing. I have already seen some of the outline fonts produced by this program and it looks very promising - there should not be the same problems as with 'staircase' or 'jagged edge' large lettering that you get with some programs.

D. JONES (cont'd)

I hope to be able to supply more pointer environment applications in the future. Although the original QL operating system is remarkably good in many respects, the pointer environment system enhances QDOS beyond all recognition and must be the future of QL software. Utilities like Easyptr II which make it easier for programmers to write in this environment should prove to be very popular.

I would like to think that I managed to create a reasonably successful new QL software company in a fairly short period, at a time when some companies had gone into decline. I have striven to give good customer service and to provide user friendly software which helps the user to achieve what he or she wants to do with the QL. I have produced new software and kept alive some software which might otherwise not be available to QL users any more. I have also, through marketing other people's software, managed to help to keep many individuals and smaller companies producing QL products.

I have tried to achieve this with a friendly, personal service, and without getting involved in any "my product is better than yours" slanging matches. By nature, I don't like upsetting people, and it is nice to be able to think that I have achieved something in business without treading on many people's toes!

I have stayed with the QL because I am an enthusiast. I basically enjoy using it and programming it, I find it suitable for my needs and I have also made many new friends through it. I hope to be around for some time yet providing support for this remarkable little computer and its dedicated users worldwide.

Dilwyn Jones

SAVE AND LOAD ARRAYS
MASSAPEQUA PARK, NEW YORK, USA - BOB MALLOY

If you have ever created an array in SuperBasic you may have run into a slight problem. How do you save it? The manual is of no help. At least I couldn't find instructions there. Even Jan Jones excellent book was, in this area, totally lacking. The program that follows will show the procedures to save and load both numeric and alphanumeric (text) arrays. The program itself is totally useless, but HOW it does it, may be of help to those of you who were accustomed to working with arrays on previous Sinclair computers.

When it runs, the program will create two arrays, **number** and **NAME$**. Each can contain five elements. The entries in Name$ can be up to 10 letters each. Just enter five numbers as asked and then five names. (That's the easy part).

There are four procedures in the program for saving and loading:

> savnum at line 200
> loadnum at line 300
> savname at line 510
> loadname at line 605

ARRAYS (cont'd)

As written the files will be saved to ramdisk. If you do not have ramdisk capabilities simply change the appropriate lines to mdv or flp.

To use the program after typing it; RUN and then answer the prompts. You can check an entry by typing PRINT number(3) or PRINT name$(3,1 to). NOTE: With text you must give full definitions. Typing PRINT name$(3) will get you nowhere. Now type savnum (ENTER) and savname (ENTER) to invoke the procedures. Next type CLEAR to clear out the variables. Typing loadnum (ENTER) and loadname (ENTER) will bring your variables back.

```
100 DIM number(5)
110 FOR c = 1 TO 5
120 INPUT "number ";number(c)
130 NEXT c
200 DEFine PROCedure savnum
210 REMark save number
220 OPEN_NEW #3,ram1_nu
230 FOR c = 1 TO 5
240 PRINT #3,number(c)
250 NEXT c
260 CLOSE #3
270 END DEFine
300 DEFine PROCedure loadnum
310 REMark load number
320 DIM number(5)
330 OPEN_IN #3,ram1_nu
340 FOR c = 1 TO 5
350 INPUT #3,number(c)
360 NEXT c
370 CLOSE #3
380 END DEFine
400 REMark text
410 DIM name$(5,10)
420 FOR c = 1 TO 5

430 INPUT "NAMES ",name$(c)
440 NEXT c
500 REMark SAVE NAME$
510 DEFine PROCedure savname
520 OPEN_NEW #3, ram1_nam
530 FOR a = 1 TO 5
540 FOR b = 1 TO 10
550 PRINT #3,name$(a,b)
560 NEXT b
570 NEXT a
590 CLOSE #3
595 END DEFine
600 REMark load name$
605 DEFine PROCedure loadname
607 DIM name$(5,10)
610 OPEN_IN #3, ram1_nam
620 FOR a = 1 TO 5
630 FOR b = 1 TO 10
640 INPUT #3, name$(a,b)
650 NEXT b
660 NEXT a
670 CLOSE #3
680 END DEFine
```

QL ROM SWITCHING BOARD - REVIEW
BANGOR, GWYNEDD, GREAT BRITAIN - DILWYN JONES

The QL ROM Switching Board is a small circuit board, produced by members of the QLEA Quanta sub-group in England. This board allows you to install both a Minerva EPROM plus an original Sinclair ROM into the QL. Either of the two ROMS can be selected by means of a switch and a reset of the QL. The selected ROM is indicated by a two colour LED on the top of the QL case, green for a Sinclair ROM, red for a Minerva.

Having this device installed means you can now use your Minerva EPROM as your standard QL operating system without fear that you will lose the use of the few programs which fail to run properly under Minerva. Now you can effortlessly revert to the old ROMs at the flick of a switch just for running those few problematical programs.

ROM SWITCH (cont'd)

The exact opposite is also possible. Software writers can program on a version JM or JS (or national variants) ROM machine, and flick to Minerva to check for compatibility on that system.

So now you can have the best of both worlds - you can use the Minerva system as your main operating system, but you can also fall back to the older versions of the ROM when you run into problems with programs which cannot be made to run on a Minerva system.

Currently, this device only works with the mark 1 Minerva, but QLEA tell me that at the time of this writing, they are working closely with TF Services (who sell Minerva) to produce a version which will work with the more recent mark 2 Minerva.

The QL ROM Switching Board works with a QIMI mouse interface and with a Gold Card, including the four drive adaptor card. Apart from the mark 2 Minerva, I have not found that it fails to work with any QL add-ons except internally mounted add-ons which obstruct the PCB where it installs into place over the ROM sockets. As far as I can determine, it takes no addressing space in the QL at all.

Installation is as simple as a device of this type can be. There are one or two slightly tricky steps, such as drilling two holes in the case for the indicator light and the selector switch for the ROMs, but it is mostly a plug in and go system.

Simply remove the old ROM chips from the QL, and put them on the new carrier printed circuit board, add the Minerva EPROM (but you will normally have to bend one of the legs on this chip), then plug it all onto the QL circuit board as indicated in the instructions sheet. This sheet shows a diagram, has clear instructions and also a couple of telephone numbers where you can get help if something goes wrong. If you prefer, QLEA will fit it for a minimal fee, but this is really onlypractical for British users since you have to send the QL to them or wait until you meet them at a show. In any case, it would be best to contact Geraint Jones first (address and telephone number below).

You need to lay out the LED and switch of course to reach the holes in the case. The switch is probably the only aspect of this device which I can find fault with, since it protrudes from the back of the QL and can be operated accidentally if something brushes against it. This has happened to me once or twice, (I keep my most used disks in a row behind the QL, resting against the side of the disk drive and I have managed to knock a disk against the switch). When time permits, I intend to correct this by fixing pieces of wood or plastic either side of the switch to protect it against accidental operation. Accidental operation of the switch will probably crash most QLs, but since a Gold Card copies all of the ROM into RAM at reset time, this may not have any significant effect on a Gold Card system. However, don't rely on that!

So far, this cute little device has been nothing but useful to me. It has made one of my QLs redundant as I used to keep three QL's with different ROM versions for software compatibility testing. It has not interfered with anything installed (software or hardware), and has already proved its value in the few weeks I have had it installed.

The device is highly recommended to existing or aspiring Minerva owners and software writers, and I found it was worth every penny of the purchase price.

ROM SWITCH (cont'd)

PRICE: £25.00 (plus £5.00 fitting charge unless you fit it yourself)
Postage free in the UK, £3.00 to other countries.
(NB £ is UK Pounds Sterling)

SUPPLIER: QLEA (QL East Anglia)
c/o Chris Howard
13 Oak Grove
Horsford, Norwich
Norfolk
NR10 3DR
Great Britain.

Contact: Chris Howard (0603) 891183

STATISTICAL ANALYSIS SYSTEM
DUBUQUE, IOWA, USA - MEDAC

Mechanical Design Analysis Consultants (MeDAC) has decided to market their Statistical Analysis System. This system was designed by our Chief Design Analyst (Glen E. Smelcer) after 12 years working in the Design Analysis Department of INTERNATIONAL HARVESTER CO. and 15 years working in the Reliability Design Analysis Department of the JOHN DEERE Dubuque Works.

Statistical Analysis Techniques used by these companies, the major Automotive companies, and their suppliers use random sampling statistics taught in all College and University courses. Unfortunately, their testing interest wanes long before tests past the mean are completed. Tests past the 15th percentile are rare.

The new Total Quality Management Techniques help improve manufactured quality. Infant mortality failures are now almost non-existant. If a company is honest in their laboratory and prototype testing, they will have test failures similar to what their customers can expect. These failures are more Normally Distributed than the raw data indicates.

MEDAC's system takes the test data and calculates: Pearson III statistics of mean, variance, standard deviation and coefficient of skewness in the normal way. It then translates this into the Weibull parameters of Weibull slope and characteristic life. It accounts for unfailed items in the test samples by assuming they are Normally Distributed. If the number of test data items are less than 30 samples, the system allows a choice of a Maximum Likelihood Analysis.

We would like to introduce this System in one or two day seminars. They would cover how to minimize test samples, the statistics involved and how to predict the number of Spare Parts required based on annual sales. We would provide participants with Basic computer programs on 5.25" disks usable on minimum XT personal computers. The entire system is also available for the QL on Microdrive cartridges. Please see our advertisement in this issue for additional information.

STRINGS AND THE C-PROGRAMMING LANGUAGE

EAST PROVIDENCE, RHODE ISLAND, USA - WILL HORTON

Strings are a means of storing and manipulating text, such as names and sentences. In C, strings are not a formal data type but an array of type "char". When you think about a string in C, it is a series of characters, formed into an array. In other programming languages strings are also treated as arrays but it is transparent to the programmer.

Keeping in mind that a string is merely an array of characters let us demonstrate how a string is initialized in C. The first and most demonstrative means is to perform the operation shown below. Note that this can only be done once at the beginning of a function since it declares that "name" is of type "char" and it also initializes it to the string "Hello"

```
static   char   name[] = {'H','e','l','l','o','\0'};
```

The command "static" is a storage class used to assign the values:H,e,l,l,o to the character array "name", note that each character element is a byte in length. Once the variable array has been assigned a value it cannot be reassigned in this manner. C also requires that a string be terminated with a null character. The "\0" character is referred to as the null character which is equivalent to the ASCII zero.

C realizes that strings are a special case and it has made a provision to simplify the initialization of a string.

```
static char name[] = "Hello";
```

This is a straightforward method that reduces the extra work required in the previous example. Once again this initialization is only performed once at the beginning of the function.

There was no need to dimension the array in the previous example since the length of the string was made known to the compiler by the length of the name "Hello". If you where going to assign a string to "name" at a later time you could do the following:

```
char name[32];
```

This would assign enough room in memory for a string thirty one characters long, with one character left over for the null character.

Now, how do values get assigned to a string if they are not initialized at the onset of the function? The answer to this question is either by keyboard input functions, or one of the many string manipulating functions that are part of the standard C library.

Let us start by showing how to use the keyboard inputing functions. The two most common functions are shown below.

```
scanf("%s", name);
gets(name);
```

The "scanf" function is good for loading data into more than one variable, and the variables do not have to only be strings, they could be integers or floating points. For example we

92

STRINGS (cont'd)

could write: scanf("%s %d %d",name,&age,&weight), so the "scanf" function is very versatile. Note here that "age" and "weight" have been made into a pointer by using the address operator "&", a pointer is required by this function.The variable "name" did not need this since it is already an address. There is one problem with using the "scanf" function and that is, it uses any whitespace to determine the end of a variable. If you enter a multiword string input it would only see the first string. The solution is to use the "gets" function if you have a multiword string input. But remember you can only enter one string.

By handling strings as arrays a great deal of flexibility can be gained as you begin to work with strings. For example consider the following listing:

```
#include <string.h>
#include <stdio.h>
main()
{
        static char member[] = "John Doe";

        printf("%s\n",&member[5]);
        printf("%c\n",member[5]);
}
```

The first printf statement would print "Doe", since the array is pointing to the sixth character in the string. The address operator "&" is what tells the "printf" statement where to begin printing. Note that a string begins with "0" as its first character location and goes to length of string minus one. So for string "John Doe", the character "D" in "Doe" is at location five because the counting starts at zero and not one.

The second "printf" statement merely prints the first character in the name "Doe" or "D". This is so since the string "member" is set at character position six, and by using the format specifier "%c", it tells the "printf" function to only print a character. The format specifier "%s" indicates to the "printf" function to print a string.

Pointers and Strings.

There is a connection between strings and pointers, because strings are arrays and arrays are closely related to pointers. So a string can be initialized as a pointer, let us compare the previously learned method of initializing a string with the new pointer method.

static char message[] = "Hello There"; /* previous method */

char *message = "Hello There"; /* pointer method */

Both methods set aside an array with enough space for eleven bytes plus one byte for the '\0'(null) character. For the array method the address of the first character is given the name of the array, in this case "message". For the pointer version an array is set aside in the same way, with the addition of a pointer variable. It is this pointer variable that is given the name "message". The figure below shows how both methods are setup in memory. An arbitrary address sequence is used to give the reader an idea of how the string looks in memory.

STRINGS (cont'd)

message[] = "HELLO THERE"

message is a pointer constant

*message = "HELLO THERE"

message is a pointer variable
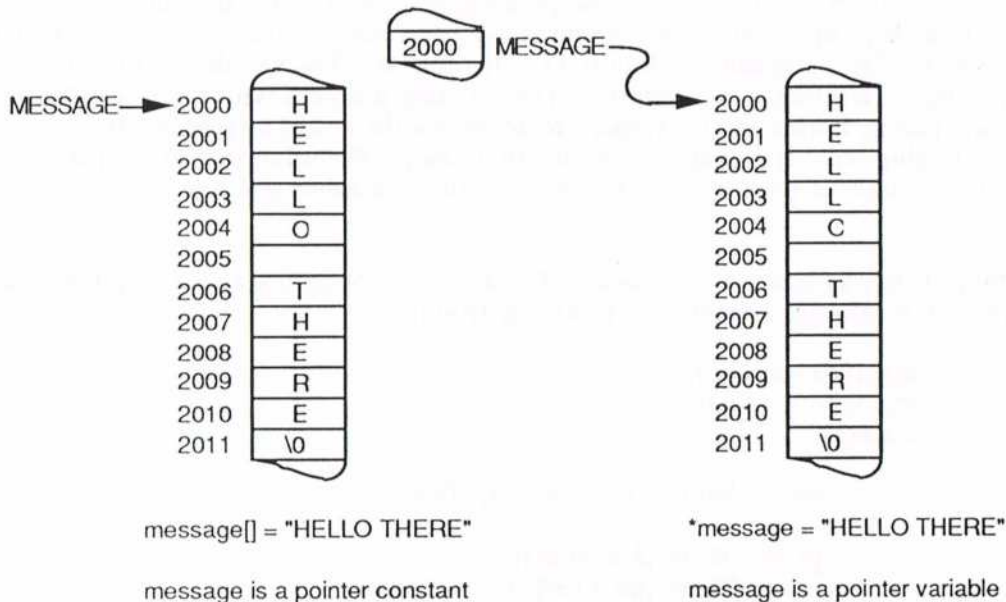
STRING ARRAY AND POINTER ARRAY MEMORY ALLOCATION

The following program shows how a string initialized as a pointer can be manipulated.

```
/* Using the pointer method */
#include <string.h>
#include <stdio.h>
main()
{
        char *message = "Hello There";
        /* increment the pointer six times */
        for(n=1;n<7;n++)
                ++message;
        printf("%s\n",message);
        printf("%c\n",*message);
}
```

The pointer method gives the same results as the array method but this time we are manipulating the pointer "message" which is pointing to the array "Hello There", located somewhere in memory. The first "printf" statement will print "There", this is so because the pointer "message" has been incremented six times to point to the seventh character of the string "Hello There" it will then start printing from that point.

The next "printf" statement only prints the character "T", because the indirection operator "*" has been added in front of "message", and the format specifier "%c" is used in the "printf" function. The indirection operator tells the "printf" function to look at the contents of what the pointer "message" is pointing to, which is the character "T", and print it through the direction of the format specifier "%c", which denotes a character.

94

STRINGS (cont'd)

Another application of the pointer method is in the manipulation of strings. If we have a list of items that need to be sorted into alphabetical order we can setup an array of pointers to strings. This creates a convenient means of manipulating the pointers to reorder the strings into alphabetical order. Refer to the listing below which shows an array of pointers to strings called "*fruit[5]". This sets up five pointers that point to the five fruits that are listed in the program. Compare this to the way in which a string is initialized, and now you have an array of five strings being initialized.

In order to sort this array of pointers a simple bubble sort is used which requires only the swapping of pointers to put the array into order. The "strcmp" function is used to test the two strings, and a swap occurs when the first string is greater than the second string. Keep in mind that the five strings occupy locations in memory that do not move. It is the those locations in memory(of the fruit names) that the pointers "fruit" point to that are being juggled around not the locations of the names of the fruit.

```c
#include <stdio.h>
#include <string.h>
main()
{
        char *temp;
        int in, out;
        /* initialize pointers to strings */
        static  char  *fruit[5] = {  "Peach" ,
                                "Orange" ,
                                "Apple" ,
                                "Pear"  ,
                                "Kiwi"   };


        /* sort the list into alphabetical order using the
           string compare function "strcmp"            */

        for(out=0; out<4; out++)
            for(in=out+1; in<5; in++)
                if (strcmp(fruit[out],fruit[in]) > 0 )
                {
                        /* swap the pointers */
                        temp = fruit[in];
                        fruit[in] = fruit[out];
                        fruit[out] = temp;
                }

        /* Print the sorted strings */
        for(out=0; out<5; out++)
            printf("%s\n",fruit[out]);

}
```

The first thing this program does is to assign the strings "Peach", "Orange", "Apple", "Pear", and "Kiwi" to the string pointer "fruit". The five fruits are now located somewhere in memory and pointed to by the pointer array "fruit". To sort the fruit into alphabetical

STRINGS (cont'd)

order the bubble-sort is set up with two loops, an outer loop and an inner loop. When a compare takes place using the "strcmp" function the pointers will be swapped if the alphabetical order of "fruit[out]" is greater than "fruit[in]". This simply means that what is being pointed to by "fruit[out]" will now be pointed to by "fruit[in]" and vice versa. Note that the stings do not move but the pointers that point to them are reordered. The sort will continue until all of the strings are in alphabetical order.

To make the handling of strings easier, the C-Programming Language offers a number of functions to manipulate strings. The following is a list of some standard functions, remember to include the header <string.c> when using these functions in your programs, or they will not compile.

gets()
    char name[81];
    gets(name);
    Reads a string from the keyboard and assigns it to "name".
    Not as versatile a scanf(), but is able to read string that
    contains white space. This can not be done easily with
    scanf().

puts()
    puts("Hello There");
    This is a special function for putting a string to the screen.
    It can only handle one string at a time.

scanf()
    scanf("d %c %f",&event, &heat, &time);
    An input function that is very versatile at inputing data
    from the keyboard.

strcat()
    strcat(string1,string2);
    Concatenates string1 to string2.

strchr()
    ptr = strchr(str, 'x');
    This returns a pointer to the first occurrence of the character
    'x' in the string 'str'. This is not the position in the string
    from 0, but the address from where the string starts.

strcmp()
    r = strcmp(string1,string2);
    This compares two strings and returns an integer value based
    on the comparison. /* < 0 string1 < string2
                 = 0 string1 == string2
                 > 0 string1 > string2 */

strcpy()
    static char string1[] = "cart";
    static char string2[] = "cat";
    strcpy(&string1[3], &string2[2]);
    Copies "string2" into "string1". In this example this will
    start at position 3 of "string1" and position 2 of "string2".

strlen()
    x = strlen(name);
    This returns the length of the string "name".

SCREEN SNATCHER
MASSAPEQUA, NEW YORK, USA - BOB GILDER

Dilwin Jones Computing

Screen Snatcher is a powerful utility which has one purpose, to save a copy of the current screen display to disk, microdrive or RAM disk. This program may not appear to be of value to everyone using a QL, however, if you have ever wanted to save a copy of text mixed with some foreign characters while using QUILL or any other program, you will find that with the exception of Screen Snatcher, it can be extremely difficult and often times impossible to activate screen dump utilities. Why would you want to save a file from QUILL as a screen? More on this later.

Screen Snatcher will operate on any QL, from a standard 128K version up to the Two Meg Gold Card machine. The program, when loaded resides in memory alongside of your other programs and multi-tasks well.

My copy of Screen Snatcher arrived on disk (copies on microdrive cartridge are also available), and was accompanied by an eight page manual providing clear instructions on program requirements, program purpose, why the program was written, making a backup, starting the program, and numerous other topics. I glanced through the documentation and decided to load the program (after making a backup) to see how the program worked. A BOOT program is included, or Screen Snatcher can be loaded using 'EXEC flp1_SNATCH_obj'.

When the program is loaded a menu appears on the screen which allows the user to make changes to any of the default settings:

0	Snatch activate key	: CTRL + s
1	Filename for saved screen	: flp1_SNATCHED_scr
2	Key to abort screen snatch	: CTRL + q
3	Screen address	: standard (hex 20000)
4	Activate screen snatcher	:

Up/Dn cursor keys=List F1=HELP F4=REDRAW ESC=QUIT

PRESS "SPACE" OR "ENTER" TO SELECT MENU ITEM

Please note that item 3, Screen address should be left as is unless you are using 'Minerva' and know the address of screen two.

Use of the up/down cursor keys allows placement of a highlighted bar to be used for selection of menu items 0 to 5. If you use Turbo QUILL then I would recommend that you change item '0, the Snatch activate key key combination **CTRL + s**' to another character as Turbo QUILL looks for a start-up file when loaded which is setup with **CTRL + s** as default!

Making a change is simple. Place the white bar over the item you wish to change and press SPACE or ENTER. Press CTRL and whatever character you wish to use as the Snatch Activate key and it will appear on the menu. Since I use Turbo QUILL,I changed the Snatch Activate key to **CTRL+ a**.

You may want to change the default filename to something suitable to the file contents, and if you intend to save more than one file from the same or modified file you may consider

SNATCHER (cont'd)

adding a number as the last character of the name such as QUILL1_scr. The drivename can also be changed if you wish to use 'flp2_ or mdv1_'or any other media device.

I left the other items on the menu as is, and then selected item 4 on the menu, to 'Activate screen snatcher'. I then loaded QUILL with the command "EXEC_W flp1_QUILL". Within seconds the familiar QUILL screen appeared and I began typing in some text. I then pressed **CTRL+ a** and the drive light illuminated and the screen from QUILL was saved to disk.

Now, why would I want to save a screen from QUILL when all I have to do is to save the file as a document? The snatched screen is saved in a format that can be directly loaded into most QL Drawing and DeskTop Publishing programs. You can load your saved screens and edit them to compliment your artwork or use the artwork to complement your text.

Have you ever wanted to have a print-out of your printer translates after a session with INSTALL_bas? Have you ever used the SEDIT command in ARCHIVE to design a special screen, worked on it awhile, made some changes and then didn't like what was now on the screen and wished you could see what your original screen looked like? I am sure we've have all had these moments!

When you have saved a screen with Screen Snatcher, and want to view the file, enter the command 'LBYTES drivename_filename,131072'. The screen will immediately appear.

If you run into difficulty remembering which sequence is used for 'snatching', just press F1 and a small HELP file will come to your rescue.

How do I rate Screen Snatcher? On a scale of 1 to 10, a big TEN!

Editor's Note: Refer to the Dilwyn Jones Computing advert in this issue for ordering information. At £10 SCREEN SNATCHER is a bargin.

NEWS FLASH!!!!

In support of a rapidly expanding hardware line, Miracle Systems welcomes Noud Sneider, a Dutch engineer and QL enthusiast to their staff. The basis for the next generation QL will center around the new graphics card under development. All other hardware expansions will plug into this card. We hope to see the Graphics Card shipped by Christmas. More when available.

In addition, Jochen Merz is ready to release version 1.04 of QSPREAD, his pointer environment spreadsheet. This update should be available by the time you read this.

Tonkin's First Computer Dictionary

Documentation: (n.) a novel sold with software, designed to entertain the operator during episodes of bugs or gliches.

Processor: (n.) a device for converting sense to nonsense at the speed of electricity, or (rarely) the reverse.

QL SERIAL CABLE PINOUTS
MASSAPEQUA, NEW YORK, USA - BOB GILDER

The following diagrams illustrate the pinouts for the two connectors required for assembling a serial to printer/modem cable. The parts are available at any Radio Shack (Tandy) store.

13 12 11 10 9 8 7 6 5 4 3 2 1
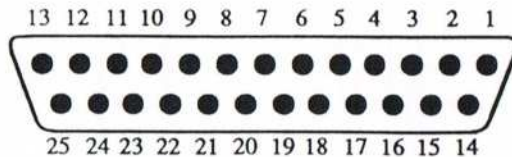
25 24 23 22 21 20 19 18 17 16 15 14
25 Pin male 'D' Connector
(Solder pin side)

5 4 3 2 1
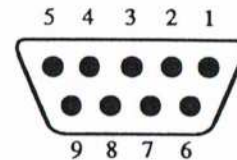
9 8 7 6
9 Pin male 'D' Connector
(Solder pin side)

CONNECTION TABLE

25 Pin 'D' Connector - To - 9 Pin 'D' Connector

Pin numbers   Pin numbers

* Black	#2	←——————→	#2	TXD
* Brown	#3	←——————→	#3	RXD
* Orange	#5	←——————→	#5	CTS
* White	#6	←——————→	#9	DSR
* Green	#7	←——————→	#7	SG
* Blue	#20	←——————→	#4	DTR

* Wire colors of original QL Serial cable

TXD = Transmitted Data
RXD = Received Data
CTS = Clear To Send
DSR = Data Set Ready
DTR = Data Terminal Ready
SG = Signal Ground

13 12 11 10 9 8 7 6 5 4 3 2 1

25 24 23 22 21 20 19 18 17 16 15 14

5 4 3 2 1

9 8 7 6
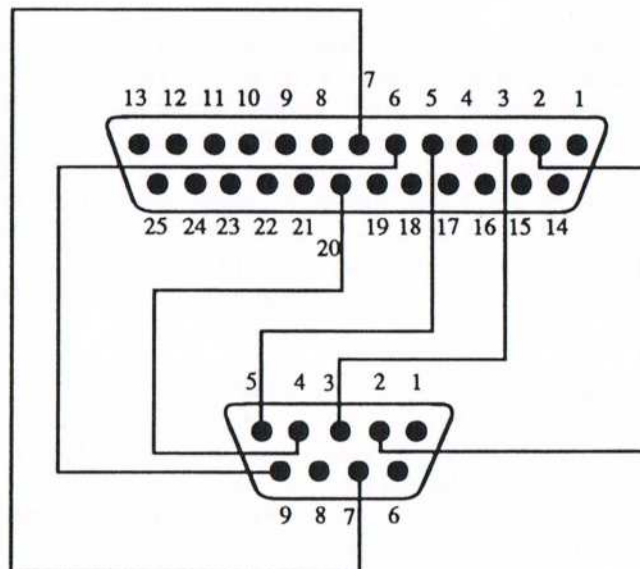
SERIAL CABLE SCHEMATIC DIAGRAM

NOTE: Connectors are shown from the solder pin side

SERIAL CABLE (cont'd)

PARTS REQUIRED

Quantity	Item	Part #	Price
1	9 pin 'D' male connector	276-1537	$ .99
1	25 pin 'D' male connector	276-1547	$ 1.47
1	9 pin connetor hood	276-1539	$ .79
1	25 pin connector hood	276-1549	$ .89
6 ft	(more or less) 9 conductor cable (only 6 wires used)	278-775	$ .59 per ft.

NOTE: All part numbers listed are Radio Shack numbers, however these are common parts and should be available at any electronics store.

DIY PRINTER DRIVER
ST JOHN'S, NEWFOUNDLAND, CANADA - HOWARD CLASE

Taylor Penrose's program and challenge in the July/August 92 issue of IQLR set me thinking. I originally intended to address his point #4, how to speed it up, but in the process I seem to have dealt with most of his other points too. The result is given in the listing CGP_220_Driver, it isn't just a type-it-in-and-RUN program, but a prototype that you can turn into a supplementary custom printer driver for more or less any purpose and any printer, although for demonstration purposes it includes his data for the CGP 220.

I've also supplied the listing of the experimental program that evolved as I worked out my ideas. There are several tricks I used in it that readers might be interested in.

My first reaction was to the 64 very similar lines in the SELect loop; in my experience when you find yourself typing in lots of similar lines there is almost always a better way, and while driving home that day (I had read the article during my lunch break.) I realized what it was here: use an array as a "look up table". Once Taylor's program has entered the SELect loop it has to check about 30 values on average before it finds what it is looking for, while since there is a 1:1 correlation between initial and final value of t it can go straight to the right answer in an array. So the first thing I did was to type up a set of DATA statements with the final 64 values for t (I had to include the "no change" values too of course, it's much the simplest and quickest way.) and read them into an integer array (120).

I then added a few more lines (200 - 235) to create a PROCedure that mimicked Taylor's. I thought it would be faster, but how to test it without typing in all those SELect conditions? Who do I usually get to do all my boring repetitive tasks? No, no sexism please, I mean my QL of course! Looking at the listing I saw there was a regular pattern to everything but the final values, and I already had those in an array, so I could write a PROCedure to create a file containing the necessary lines, and MERGE them into the program. This is called Gen_lines in the Test_converts listing. You cannot use some keywords within PROCedures or FUNCtions, and MERGE is one of them, so the MERGEing has to be done from the keyboard. The results can be seen in lines 320 to 395, note that it is possible to have more than one condition per line, your programs don't have to look like the snakes editors hate! I've altered Taylor's routine slightly to conform to SuperBasic structure

PRINTER DRIVER (cont'd)

conventions and eliminate checking of values over 191 (unnecessary if these don't occur!) this gives a slight increase in speed and added the missing END DEFine statement. In order to test the routines I converted them to FUNCtions so that I could check that they both gave the same results (45); test$ is just a list of all 256 possible characters it is converted first into a$ by my routine and then b$ by Taylor's, timing each operation.

On my GoldCard/Minerva combination this was so quick I had to put in some loops to get meaningful timings. My array routine was about four times faster than Taylor's SELect one on this string. You can also speed his up about 10% by omitting all mention of codes for which no change is required.

A couple of things to note:

1. The double brackets around test$ in the FUNCtion calls. The first set is because it is a FUNCtion, the second prevents test$ from being changed by the routines; only the "value" is passed in this case so is is not treated as the formal parameter within the FUNCtion. (Try the effect of omitting one set!)

2. that I used SELect (220) rather than the alternative, "IF t>127 AND t<192:"; SELect is noticeably faster in this case in my system. (Try it on yours!)

If you compare the experimental listing with the final version you will see that I have made a few changes. The major one is that it now works on whole files rather than just strings, and sends its output straight to the printer. The thinking behind this is that since it is rather slow anyway it makes sense to couple it to that other slow step - the actual printing when you probably won't notice it. If you were working in Quill then you would write your document in the usual way, but print it to a file instead of directly to the printer. To do this just type in a filename when the "printer" prompt appears, the output is redirected to a file called filename_lis on your default device, but it still goes through the Quill printer driver, so all the normal printer controls will have been added. (If you want a different device or three letter extension then type them in as well). If you then run it through my supplementary printer driver it will, in addition, change the characters with codes between 128 and 191 to whatever you wish as it prints the file. (This is more or less the way that the original printer driver works.)

The other major difference is that instead of writing the codes into an integer array I have converted it to a string array. My original idea here was that the integer to character conversion had to be done once only for each code instead of every time it occured in the file.

In fact this doesn't seem to save a significant amount of time, as I had hoped; but it does make it easy to substitute a string of any number of characters for a single one in the original file. So there should be no problem in sending the code to turn on graphics, print your graphics character, and switch back to normal printing; but not having a CGP 220 myself I shall have to leave the user to work out the required codes.

As an example, in line 145 I have included the seven character code for printing the "`" sign correctly to my Epson printer. Note that this can be a mixture of string variables (esc$), CHR$() functions, and actual characters (in quotes of course). This will overwrite whatever value has been read in from the DATA list. You may also have to change the

PRINTER DRIVER (cont'd)

DIM statement in line 130, the first number is the total number of strings 64 for all the characters form 128 to 191, the second parameter should be one less than the maximum length of the longest replacement string you want to use (QDOS counts from zero here!).

You should also OPEN#4 in line 125 to whatever your own printer requires (mine is actually ser2hc). If the program crashes for any reason then type "Finish" at the keyboard to close the open channels, or just "CLOSE" if you are using TK2.

With a little imagination this program could be adapted to act as a supplementary printer driver between any word processor output and any printer, or the output from a SuperBasic program, as long as it is written to a file first.
(c) 1992 H.J.Clase

```
5    REMark  CGP 220 driver
10   REMark (pd)  1992.07.03 hjclase ver 1.0
15    REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^Main
20   Setup: FPrint: Finish: STOP
25    REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
100 DEFine PROCedure                              Setup
105 LOCal n%,n$,esc$: esc$=CHR$(27)
110 INPUT"Device and file name to print:"\n$
115 REMark Edit the next 2 lines as required<<<<<<<<<<<<<<<<
120 c%=3: OPEN_IN#c%,n$:    REMark unused channels<<<<
125 OPEN#c%+1,ser1:   REMark Printer channel specs<<<<<<
130 DIM new%(64,9): RESTORE
135 FOR i=1 TO 64:READ n%(i)=CHR$(n%)
140 REMark Next line example only.<<<<<<<<<<<<<<<<<<<<
145 new$(64)=esc$&'R'&CHR$(3)&'#"&esc$&"R"&CHR$(0)
150 DATA 182,206,211,187,183,207,213,184,162,205
155 DATA 209,139,200,161,192,198,189,193,199,201
160 DATA 148,194,202,151,195,203,188,196,185,180
165 DATA 176,164,177,161,210,215,178,165,212,179
170 DATA 168,214,208,171,172,173,174,175,165,175
175 DATA 178,204,220,181,169,183,184,185,166,187
180 DATA 188,189,190,191
185 END DEFine : REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

200 DEFine PROCedure                              FPrint
205 LOCal a,loop: REPeat loop
210    IF EOF(#3):EXIT loop
215    a=CODE(INKEY$(#3,-1))
220    SELect ON a
225       = 1 TO 127,192 TO 255: PRINT#4,CHR$(a);
230       = 128 TO 191:PRINT#4,new$(a-127);
235    END SELect : END REPeat loop
240 END DEFine : REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

300 DEFine PROCedure                              Finish
305 CLOSE#c%: CLOSE#c%+1
310 END DEFine : REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

PRINTER DRIVER (cont'd)

Listing 2:

```
  5 REMark        Test_converts
 10 REMark 1992.07.02 hjc
 15 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^Main

 20 Setup
 25 test$=": FOR i=1 TO 256: test$=test$&CHR$(i)
 30 t1=DATE: FOR i= 0 TO 5: a$=Convert$((test$))
 35 t2=DATE: FOR i= 0 TO 5: b$=Cnvrt$((test$))
 40 t3=DATE: PRINT t2-t1,t3-t2
 45 REMark check results the same
 50 FOR i=1 TO 256: IF a$(i)<>b$(i): PRINT!i;
 55 STOP
 60 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

100 DEFine PROCedure                          Setup
105 CLS: DIM new%(64): RESTORE
110 FOR i=1 TO 64:READ new%(i)
115 DATA  182,206,211,187,183,207,213,184,162,205
120 DATA  209,139,200,161,192,198,189,193,199,201
125 DATA  148,194,202,151,195,203,188,196,185,180
130 DATA  176,164,177,161,210,215,178,165,212,179
135 DATA  168,214,208,171,172,173,174,175,165,175
140 DATA  178,204,220,181,169,183,184,185,166,187
145 DATA  188,189,190,191
150 END DEFine : REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

200 DEFine FuNction                     Convert$(text$)
205 LOCal L,t
210 FOR L=1 TO LEN(text$)
215    t=CODE(text$(L))
220    SELect ON t
225    = 128 TO 191: text$(L)=CHR$(new%(t-127))
230 END SELect : END FOR L: RETurn text$
235 END DEFine : REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

300 DEFine FuNction                     Cnvrt$(text$)
305 LOCal L,t: FOR L=1 TO LEN(text$)
310 t=CODE(text$(L)): IF t>127 AND t<192
315 SELect ON t
320 =128:t=182: =129:t=206: =130:t=211: =131:t=187
325 =132:t=183: =133:t=207: =134:t=213: =135:t=184
330 =136:t=162: =137:t=205: =138:t=209: =139:t=139
335 =140:t=200: =141:t=161: =142:t=192: =143:t=198
340 =144:t=189: =145:t=193: =146:t=199: =147:t=201
345 =148:t=148: =149:t=194: =150:t=202: =151:t=151
350 =152:t=195: =153:t=203: =154:t=188: =155:t=196
355 =156:t=185: =157:t=180: =158:t=176: =159:t=164
360 =160:t=177: =161:t=161: =162:t=210: =163:t=215
```

```
365 =164:t=178: =165:t=165: =166:t=212: =167:t=179
370 =168:t=168: =169:t=214: =170:t=208: =171:t=171
375 =172:t=172: =173:t=173: =174:t=174: =175:t=175
380 =176:t=165: =177:t=175: =178:t=178: =179:t=204
385 =180:t=220: =181:t=181: =182:t=169: =183:t=183
390 =184:t=184: =185:t=185: =186:t=166: =187:t=187
395 =188:t=188: =189:t=189: =190:t=190: =191:t=191
400 END IF :END SELect :text$(L)=CHR$(t):END FOR L
405 RETurn text$: END DEFine :REMark ^^^^^^^^^^^^^^^^^
```

```
1000 DEFine PROCedure                    Gen_Lines
1010 LOCal i,j
1020 OPEN_NEW#3,ram1_temp
1030 FOR i=0 TO 15
1040   PRINT#3,(64+i)*5;
1050   FOR j=i*4+128 TO i*4+131
1060     PRINT#3," =";j;":t=";new%(j-127);
1070     IF (j+1) MOD 4: PRINT#3,":";
1080 NEXT j: PRINT#3: END FOR j: END FOR i
1090 CLOSE#3
1100 END DEFine:   REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

3D TERRAIN

C.G.H. Services

MASSAPEQUA PARK, NEW YORK, USA - BOB MALLOY

3D Terrain, written by Ian Thompson and Rich Mellor, is a well thought out, addictive program, that takes data from Abacus and presents it in a 3 Dimensional form *ED Note: Unlike some utility programs that generate 3D bar graphs or pie charts, this program produces a topigraphical 3 dimensional landscape presentation of the data)..* The results can be sent to a printer, using the GPRINT_PRT file copied from Easel. The program, designed primarily to provide a 3D view of topography, is quite easy to use. Numeric data is first entered into Abacus and then exported to a file for Easel. This file is then imported into 3D Terrain where you are given a graphical representation of the spreadsheet data.

I found the manual, which is provided on the disk, to be easy to understand. It includes not only instructions for using the program itself but also gives very good instruction on preparing the data in Abacus.

Several specimen files are included on the disk. These can either be loaded into the program for viewing or imported into Abacus so that the user can see exactly how data is to be prepared.

The program can be multi-tasked. I used it with Taskmaster and found it extremely easy to prepare data in Abacus, export it to a ramdisk and then load it into 3D Terrain.. (Incidentally, if you do switch back and forth between programs, pressing any key when you return to 3D Terrain will restore the screen.

3D TERRAIN (cont'd)

Because the data is prepared in Abacus it can easily be changed or manipulated. Try producing one set of data, say a pyramid, and then use COPY to replicate your original data in the adjoining areas.

According to the manual the program works with Lightning and Speedscreen, with Lightning giving the best results. This program, available on microdrive or disk, requires some memory expansion (192K minimum).

Drawbacks:

The program appears unable to recognize decimal points or minus signs, i.e. -15, 1.5 and 15 will all be treated as 15.

All in all, the program works quite well and does what it claims to do. Considering some programs I have seen, this is a real plus. If you have need for this type of program then you will find 3D Terrain to be quite useful. If, on the other hand you buy it without any specific need you will undoubtedly spend many an hour creating different and various designs. It is quite addictive!

Editor's Note: The cost of **3D-TERRAIN** is £12.50 plus postage and is available from:

> C.G.H. SERVICES
> Cwm Gwen Hall
> Pencader, Dyfed, Cymru
> Wales SA39 9HA   Tel: 0559 384 574

Tonkin's First Computer Dictionary

CPU: (n.) acronym for Central Purging Unit. A device which disgards or distorts data sent to it, sometimes returning more data and sometimes merely over-heating.

Data: (n.) raw information, esp. that supplied to the central purging unit for tranformation and disposal.

READING MSDOS DISKS ON THE QL - Conclusion
PART III: The Boot Record and File Allocation Table
OAK RIDGE, TENNESSEE, USA - MEL LAVERNE

Parts I and II implicitly assumed a 40-track disk being used on an 80-track drive. Examining the boot record will be primarily useful in determing whether this is, in fact, the case. All the MSDOS disks I have so far examined have been the 40_track (720 sector) type but others are possible. In fact, there are six "standard" floppy-disk formats in the IBM PC zoo, covering the range from 320 to 2400 total sectors per disk.

The boot record occupies one sector at the very beginning of the disk, sector 0 on side 0 of track 0. Only the first 30 bytes will concern us here, the remainder of the sector being taken up with such matters as a short program to load the operating system, etc.

MIRACLE

16MHZ 68000 HIGH SPEED PROCESSOR

DD, HD, ED, DISK INTERFACE

2M BYTE RAM

CUSTOM LOGIC CHIP

BATTERY BACKED CLOCK
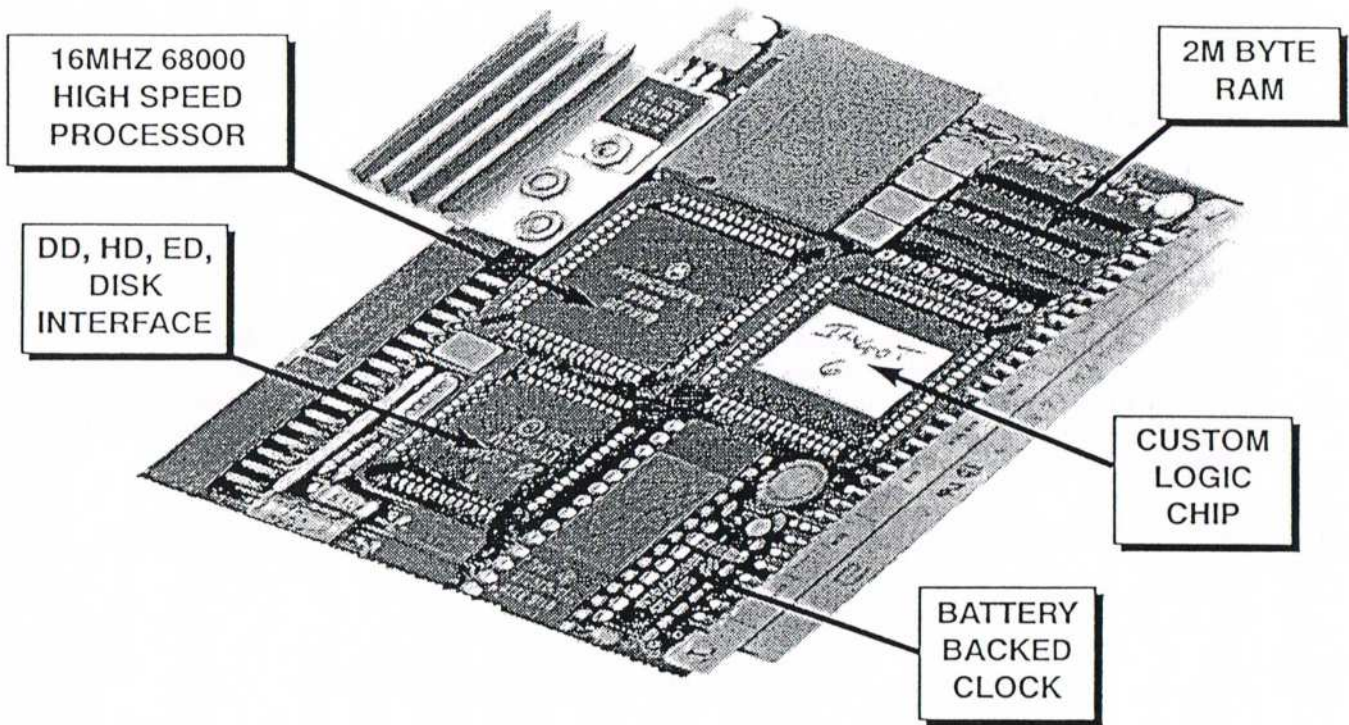
QL GOLD CARD
£225 inc. (£200 export)

This is the expansion that has been revolutionising the QL. It is very easy to fit - it simply plugs into the expansion port at the left hand of the QL - and once fitted it will instantly increase the execution speed of the QL by about 4 times due to the presence of a 16MHz 68000 on board. There is 2M of fast 16 bit RAM of which QDOS sees a contiguous 1920K. The remainder is used for shadowing the QL's ROM and display memory and for the GOLD CARD's own code.

There is a disk interface which can access 3 mechanisms (4 with the DISK ADAPTER) of 3 different densities, DD (double density, 720K), HD (high density, 1.44M) and ED (extra high density, 3.2M) in any mix. The disk interface connector is the same type that was fitted to the TRUMP CARD so most QL compatible disk drives can be used. Please note that DD drives still give a capacity of 720K per diskette. Our DUAL ED DISK DRIVE allows the GOLD CARD to access DD, HD and ED diskettes.

Another feature is the battery backed clock. When the QL is switched on the contents of the clock are copied into the QL's clock so that the time and date are correct. The firmware in the ROM gives the GOLD CARD all the functionality of the TRUMP CARD like TOOLKIT II and there is a sub-directory system for floppy and RAM disks.

Physically the GOLD CARD is about half the size of the TRUMP CARD and so fits almost all within the QL. Its current consumption is well under the allowable maximum so no special power supply is required. The GOLD CARD comes with a 14 day money back guarantee and a 2 year warranty.

SYSTEMS
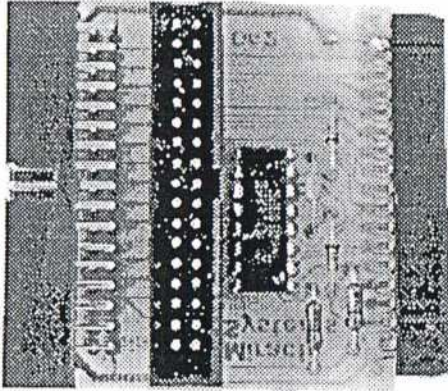
DISK ADAPTER
£15 inc. (£15 export)
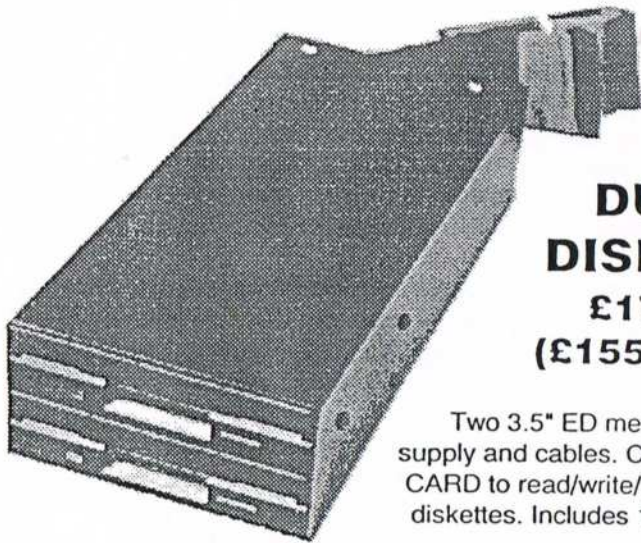
Plugs into TRUMP CARD or GOLD CARD to allow access to 2 dual disk drives (i.e. 4 mechanisms) as FLP1_, FLP2_, FLP3_, FLP4_.

BOX OF 10 ED DISKETTES
£30 inc. (£30 export)

Ten 3.5" ED diskettes. Gives capacity of 3.2 MBytes when used with GOLD CARD and DUAL ED DISK DRIVE.

QL CENTRONICS
£25 inc. (£25 export)

Connects SER1/SER2 to Centronics parallel printer. 3 metre cable included. (D9 serial plug option for German and American QL available.)

DUAL ED DISK DRIVE
£175 inc. (£155 export)

Two 3.5" ED mechanisms, power supply and cables. Connects with GOLD CARD to read/write/format DD/HD/ED diskettes. Includes 10 ED diskettes.
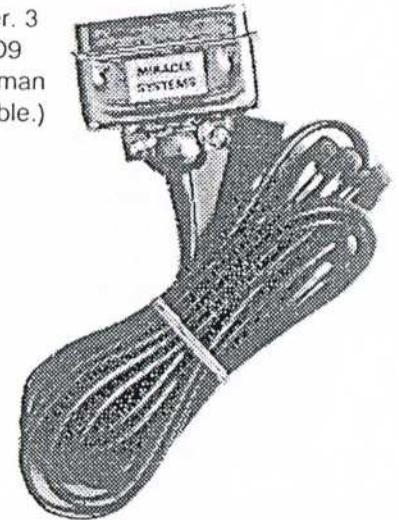
Tel: (0904) 423986

To place an order by phone please have your credit card ready. For overseas customers we charge the prices shown in brackets.

To order by post, please fill in the form opposite or write to us quoting your credit card number and expiry date, or enclosing a cheque payable to MIRACLE SYSTEMS Ltd.

To: MIRACLE SYSTEMS Ltd, 25 Broughton Way, Osbaldwick, York, YO1 3BG, U.K. Tel: (0904) 423986

Please send me _____

I enclose a cheque to the value of: £ _____

Or debit credit card ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expires ☐☐ / ☐☐

Name _____ Signature _____

Address _____

MS-DOS (cont'd)

Table 2, below, summarizes the information in the boot record that we may find useful (Table 1 appeared in Part II). Although I've given some examples of values, keep in mind that the actual entries will vary from one format to another.

| Item Description	Byte Nos	Length	Example Values
Start of boot code	2	1	$2E
System ID	4 - 11	8	MSDOS5.0
Bytes/Sector	12 - 13	2	512
Sectors/Cluster	14	1	2
Reserved sectors at beginning	15 - 16	2	1
Copies of FAT	17	1	2
Root directory entries	18 - 19	2	112
Total sectors on disk	20 - 21	2	360
Format ID	22	1	$FD
Sectors/FAT	23 - 24	2	2
Sectors/Track	25 - 26	2	9
Sides	27 - 28	2	2
Number of special reserved sectors	29 - 30	2	0

Table 2. Boot Record Parameters.

The file allocation table (FAT) follows immediately after the boot record. Normally, two identical copies are provided, the second copy simply serving as backup in case of damage to the first copy. This program uses copy 1.

The FAT shows how the disk space is used, performing essentially the same job that the allocation block does in the QL. Entries in the table generally give the "cluster number" of the next segment of a file, a "cluster" being two contiguous sectors on the disk. If an entry is zero, the corresponding cluster is unused and is available. An entry of hexadecimal $FF7 (decimal 4087) is indicative of a "bad" (i.e., unusable) cluster. The end of a listing for a particular file is shown by an entry of $FFF (4095).

On the 720-sector disk, 12 sectors are assigned to the Boot Record, the two FATs, and the Directory, leaving 708 sectors for data. With two sectors per cluster, this yields a maximum of 354 clusters. For some unexplained reason, the initial cluster is number 2, so that the maximum cluster number is 355.

Because the FAT entries are defined to be 4095 ($FFF) or less, each entry can be stored in three hex digits and a pair of entries is storable in three bytes. So far, this seems straightforward. However, because of some oddities in the 8088's data format, these bytes are stored in a somewhat convoluted form. If, for example, the three bytes were $23, $41, and $12, the two FAT entries would be $123 and $124 !

The decoding scheme is, perhaps, best shown graphically, as below:

$$23 \quad 41 \quad 12 \longrightarrow 123 \quad 124$$
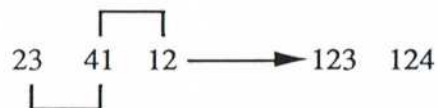
MS-DOS (cont'd)

In words, move the fourth digit to the front, the third digit to the rear. Procedure FAT uses this scheme.

The commented program follows:

```
1000 REMark File = BootFat    c M. E. LaVerne    7 Aug 92
1010 :
```

Clear the variables table, clear the screen, and select the disk drive.

```
1020 CLEAR: CLS: Get_Driv
1030 REPeat Rerun
```

Find the DOS version and volume label, if any ; select boot record or FAT for output.

```
1040   Set_VerVol vers$, vol_lbl$
1050   Menu n
```

Select screen or printer output; form$ gives a form feed for printer output or several blank lines for a separator on screen.

```
1060   out_channel ch, form$
```

Pick the boot record or file allocation table for output.

```
1070   SELect ON n
1080     = 1: Boot_Record
1090     = 2: FAT
1100   END SELect
```

If the next output is likely to overflow the page, this allows separation and dating.

```
1110   PRINT TO 10; 'Date and page this output ? (Y/N) ';
1120   m$ = INKEY$(-1): PRINT m$
1130   IF m$ == "Y" THEN PRINT #ch; DATE$; form$
```

Optional continuation. The disk may be changed at this time, if desired.

```
1140   PRINT TO 10; 'More ?'\TO 10; 'ESC to stop; any other key to continue.'
1150   m$ = INKEY$(-1): IF CODE(m$) = 27 THEN EXIT Rerun: ELSE CLS
1160 END REPeat Rerun
```

Done. Close channels; indicate completion. End of the Main program.

```
1170 CLOSE #3, #4: BEEP 10000,10: PRINT TO 10; 'Finished'
1180 :
```

Select the disk drive and open a file for direct access.

```
1190 DEFine PROCedure Get_Driv
```

```
1200  LOCal pick_drive, dr$
1210  REPeat pick_drive
1220    PRINT \TO 10; 'Drive # ? (1 or 2) ';: dr$ = INKEY$(-1): PRINT dr$
1230    IF dr$ = '1' OR dr$ = '2' THEN EXIT pick_drive
1240  END REPeat pick_drive
1250    OPEN #4, 'flp' & dr$ & '_*d2d'
1260 END DEFine Get_Driv
1270 :
```

Determine DOS version and volume label, if any.

```
1280 DEFine PROCedure Set_VerVol(vers$, vol_lbl$)
1290  LOCal a$, b$, c$, c12, i, j, p, v$
1300  GET #4\1,v$: vers$ = v$(4 TO 11): a$ = "
1310  FOR p = 6 TO 9, 257 TO 259: GET #4\p,b$: a$ = a$ & b$
1320  vol_lbl$ = '(no volume label)'
1330  FOR i = 1 TO 112
1340    j = 32*i: c$ = a$(j-31 TO j): c12 = CODE(c$(12))
1350    IF BIN(BIN$(c12,4)) > 7 THEN vol_lbl$ = c$(1 TO 11): EXIT i   1360   END
FOR i
1370 END DEFine Set_VerVol
1380 :
```

Display menu; select Boot Record or FAT.

```
1390 DEFine PROCedure Menu(n)
1400  LOCal n$
1410  REPeat Choose
1420    PRINT \\TO 10; '    MENU'
1430    PRINT TO 10; '1  Boot Record'
1440    PRINT TO 10; '2  Decode File Allocation Table'
1450    PRINT TO 10; '    Select a number from the Menu: ';
1460    n$ = INKEY$(-1): PRINT n$
1470    IF n$ = '1' OR n$ = '2' THEN
1480      n = n$: EXIT Choose
1490    ELSE
1500      PRINT TO 10; 'Invalid input = '; n$
1510    END IF
1520  END REPeat Choose
1530 END DEFine Menu
1540 :
```

Set screen or printer output. The "PAUSE 100" in line 1640 is there just to give you time to verify that the screen was indeed selected.

```
1550 DEFine PROCedure out_channel(ch, form$)
1560  LOCal out, chan$
1570  REPeat out
1580    PRINT \TO 10; 'Output to screen (S) or printer (P) ? ';
1590    chan$ = INKEY$(-1)
```

MS-DOS (cont'd)

```
1600    IF chan$ == "S" OR chan$ == "P" THEN EXIT out
1610  END REPeat out
1620  PRINT chan$
1630  IF chan$ == "S" THEN
1640    ch = 1: form$ = FILL$(CHR$(10),3): PAUSE 100: CLS
1650  ELSE
1660    ch = 3: form$ = CHR$(12): OPEN #3; prt1
```

If your printer is not Epson compatible, these codes will need to be changed. The given numbers select Elite pitch, NLQ, and margins at 10 & 94.

```
1670    BPUT #3, 27,77, 27,120,1, 27,88,10,94
1680  END IF
1690 END DEFine out_channel
1700 :
```

Outputs the boot record, as described in Table 2, to the selected channel.

```
1710 DEFine PROCedure Boot_Record
1720  LOCal a$, col: col = 28
1730  Head$ 'Boot Record': GET #4\1, a$
1740  PRINT #ch; 'System ID:'; TO col; a$(4 TO 11)
1750  PRINT #ch; 'Bytes per Sector:'; TO col; PEEKW(a$,12)
1760  PRINT #ch; 'Sectors per Cluster:'; TO col; CODE(a$(14))
1770  PRINT #ch; 'Reserved Sectors:'; TO col; PEEKW(a$,15)
1780  PRINT #ch; 'FAT Copies:'; TO col;CODE(a$(17))
1790  PRINT #ch; 'Root Directory Entries:' TO col; PEEKW(a$,18)
1800  PRINT #ch; 'Total Disk Sectors:'; TO col; PEEKW(a$,20)
1810  PRINT #ch; 'Format ID:'; TO col; HEX$(CODE(a$(22)),8)
1820  PRINT #ch; 'Sectors per FAT:'; TO col; PEEKW(a$,23)
1830  PRINT #ch; 'Sectors per Track:' TO col; PEEKW(a$,25)
1840  PRINT #ch; 'Number of sides:'; TO col; PEEKW(a$,27)
1850    PRINT #ch; 'Special Reserved Sectors:'; TO col; PEEKW(a$,29)\\ 1860 END
DEFine Boot_Record
1870 :
```

Output table headings.

```
1880 DEFine PROCedure Head$(title$)
1890  PRINT #ch\title$; ' for '; vol_lbl$; TO 50; 'DOS Version: '; vers$\\
1900 END DEFine Head$
1910 :
```

This function emulates the PEEK_W keyword. I felt this was simpler than using ALCHP, LBYTES, and PEEK_W.

```
1920 DEFine FuNction PEEKW(x$,n)
1930  RETurn CODE(x$(n)) + 256*CODE(x$(n+1))
1940 END DEFine PEEKW
1950 :
```

MS-DOS (cont'd)

Output File Allocation Table.

```
1960 DEFine PROCedure FAT
1970  LOCal sector, a$, b$, c$, d$, e$, i, j, bij
1980  Head$ 'File Allocation Table': b$ = "
```

Construct string containing copy one of the FAT.

```
1990  FOR sector = 1, 2: GET #4\1+sector, a$: b$ = b$ & a$
```

We start analyzing with byte 4, the first three bytes containing nothing useful.

```
2000  FOR i = 4 TO 1021 STEP 3
2010    c$ = "
```

Select a group of three bytes.

```
2020    FOR j = 0 TO 2: bij = CODE(b$(i+j)): c$ = c$ & HEX$(bij,8)
```

Each group is then split into two cluster numbers, using the previously described procedure, and output to the selected channel. I've left the numbers in hex for printing convenience. Printer and screen formats are the same.

```
2030    d$ = c$(4) & c$(1 TO 2): e$ = c$(5 TO 6) & c$(3)
2040    IF d$ == 'FFF' THEN PRINT #ch; d$: ELSE PRINT #ch; d$!;
2050    IF e$ == 'FFF' THEN PRINT #ch; e$: ELSE PRINT #ch; e$!;
2060  END FOR i
2070  PRINT #ch
2080 END DEFine FAT
```

Tonkin's Dictionary

Crash: (v.) to terminate a program in the usual fashion, i.e. by locking up the computer or setting a fire at the printer. (n.) the process of such termination.

Word Processor: (n.) A program which makes a $5,000 computer into a $250 typewriter. A computer game for beginning operators.

Windowing: (n., adj.) a way of making a large and easily read display into many small, cluttered, and confusing ones.

TSR: (n.) acronym for Terminate and Stay Resident. A way of turning a computer with plenty of memory into one with no memory at all.

NASA Computing

Nerheim, N – 5580 ØLEN Norway

Terms of Payment

All our products are delivered with laser printed manual in English, and the software is only available on 3.5" disks. All prices include first class AIR MAIL world wide. Allow up to 3 days for delivery.

Payment in cash or cheque. Cheques should be in NOK (not foreign currency, please) drawn on a Norwegian bank and made payable to P. Monstad.

We are regularly updating our software, and upgrades are normally free of charge. If you need more information, please write for free product information.

Disk Mate

DM is a very powerful program which enables you to manage your files and disks in an easy manner. Unlike similar programs, DM is fully menu controlled, both when selecting files and commands. File names are listed in columns which are displayed in several pages.

One of the main advantages with DM is the possibility to write a directory back to the original disk. This is very useful, e.g. after you have sorted (descending or ascending, name, type, length, update date and dataspace), changed style (upper, lower and mixed) or grouped files. The group command is the most powerful found in any QL program. It enables you to collect files into groups. Up to 26 groups can be made in one operation, and every group can contain as many files as you want. Each group can be sorted etc. individualy. When you are satisfied with the order of the file names on the screen, it is time to write the directory back. What you see on screen is what you get on disk if you use the "Write dir" command!

DM has a very powerful search-and-select routine. This routine enables you to use wildcard search, in the same manner as on the PC (? to replace an unknown character, and * to replace an unknown string). The files which then are selected (you can also select a file by selecting it manually with the cursor keys and enter or space) can be copied (very fast) deleted, renamed, printed to screen or printer, converted etc.

Only a few of the features in DM are mentioned here, and if you need more information, please write. The current version of DM (v3.20) will not work together with harddisks or level 2 sub-directories etc, but we are working on a new version which will run under the Pointer Environment. Disk Mate requires at least 512Kb of free RAM, Toolkit 2 and a disk station to run. Price: $43.50 - £22.90 - DM 69.50 - NOK 265.00

Index Optimum

A brand new version of the program previously called Index. It has now been rewritten from scratch and the benefits are enormous!

So what can it do? Well, with Index Optimum you can create a database of all your files on your disks. Index Optimum then later gives you the possibility of searching a particular file, printing directories out, sorting, updating etc. No more need to search hours through your disk collection for a specific file - Index Optimum does it in seconds! Your directory file may hold up to 1024 disks, but of course nothing stops you from creating more than one directory file. It is extremely fast to create a directory file - takes about 10 seconds per disk, included the time you spend inserting/removing disks from the disk drive. The directory file may at any time be updated, and it still works in incredible speed! Automatic, dynamic memory reservation, which means that you don't need to enter a workspace or similar - everything is carefully catered for by Index Optimum, and it doesn't capture more memory than it needs to! You may create your directory file with or without filedates. If you create it with filedates, the date when the files were last updated is also viewed on screen when searching or printing! The directory file is automatically saved after any updates and also loads automatically when necessary - no need for the user to worry about that! Reads directories of disks in about one second! Printer facilities included! Runs under Pointer Environment, i.e. mouse controlled, but can easily be controlled from keyboard! Available from August/September 1992. Please write for more info.

Price: $29.90 - £14.99 - DM 44.90 - NOK 175.00.

Dicey Business

A Football Strategy Game (Soccer) for The Sinclair QL and compatibles with at least 550Kb expanded RAM and 3.5" DD disk station. Some effects with Dicey Business:

* 4 divisions + non-league. * FA-Cup, League-Cup and European Cup. * Two games in one: Manager and Director - Choose yourself! * A-team and junior team. * Buying and selling players. * Records and statistics. * Players can get booked or sent-off during a match. * Transcriptions of tables and forms. * Sponsors. * Realistic attendances. * Economic report after each match. * The club can go bankrupt. * You may get sacked. * You may get job-offers from other clubs. * Penalty-shoot-out in the European Cup. * Injuries. * Results and tables from 4 divisions and results from all the cups. * Talent scout. * Loading and saving your team. * Over 350 Kilobytes with compiled SuperBASIC. * etc, etc...

No football skill is required to play Dicey Business. The menues in Dicey Business are very easy to use, either with the cursor keys or by direct key strokes. Dicey Business supports Epson compatible printers.

Dicey Business is supplied on disk together with a laser printed 14 pages manual.

Price: $29.90 - £14.99 - DM 44.90 - NOK 175.00.

Octal: (n.) a base-8 counting system designed so that one hand may count upon the fingers of the other. Thumbs are not used, and the index finger is reserved for the carry.

SUBSCRIPTIONS / ADVERTISEMENTS

IQLR is published by SeaCoast Services of 15 Kilburn Court, Newport, Rhode Island, USA. Our telephone number is 401-849-3805, business hours are 10:00 AM to 9:00 PM Eastern Standard Time, (1500 - 0100 GMT) Monday through Saturday.

ALL material published in IQLR remains the property of SeaCoast Services, and is COPYRIGHTED, unless otherwise specified. Written permission from SeaCoast Services IS REQUIRED for use of material printed in IQLR.

SUBSCRIPTION RATES: $14.95 per year (US FUNDS) for the US, $16.95 per year (US FUNDS) for Canada, and $28.00 per year (US FUNDS) the rest of the World. Single issue rates are $5.00 (US FUNDS). We will accept Pounds Sterling or DM bank notes (currency) equivalent to the US $ amount.

IQLR is published 4 to 6 times per year, with individual issue size determined by the amount of available material. Our subscription year is 1 May through 30 APRIL.

Mid-year subscribers will be sent back issues for the current year.

ADVERTISING RATES: $15.00 DM 25   £10   per QUARTER PAGE per issue.
                       $25.00 DM 40   £15   per HALF PAGE per issue.
                       $50.00 DM 80   £30   per FULL PAGE per issue.

ADVERTISING DEADLINE FOR THE NEXT ISSUE IS 20 OCTOBER

SUBMISSIONS DEADLINE FOR THE NEXT ISSUE IS 10 OCTOBER

IQLR WILL REVIEW OR RETURN ANY COMMERCIAL HARDWARE AND SOFTWARE SUBMITTED.

AVAILABLE FROM IQLR:

QL SURVIVOR'S SOURCE BOOK including P & H $10.95
Spiral bound VOL. 1 of IQLR including P & H $15.95
Back issues of QL WORLD $ 4.00 ea

DISCLAIMER: SeaCoast Services reserves the right to publish, edit, or reject any submitted material, and under no circumstances will SeaCoast Services be liable for any direct, indirect, or consequential damage or loss arising out of the use or inability to use, software or hardware related items published in IQLR.

GO STRAIGHT TO THE SOURCE:

Sinclair QL World

The frontline for reviews of software, hardware & literature, the best in classifieds and ads for the latest products in Europe.

Subscribe or miss out!

Sinclair QL World is published by Arcwind Ltd.

SUBSCRIPTION FORM

U.S. RATE: £40.90

Selected back issues available on request. Please contact us for details.
Cheques should be made payable Arcwind Ltd.
or
Please debit my credit card Access/Visa

Card No. _____

Expiry date _____

Please send me a 12 month subscription to Sinclair QL World

Name _____

Address _____

_____

Tel No. _____

Signed _____   Date _____

Send the completed form to

Arcwind Ltd
The Blue Barn, Tew Lane,
Wootton, Woodstock
Oxon OX 7 1HA
England
Tel: 0993 811181
Fax: 0993 811481